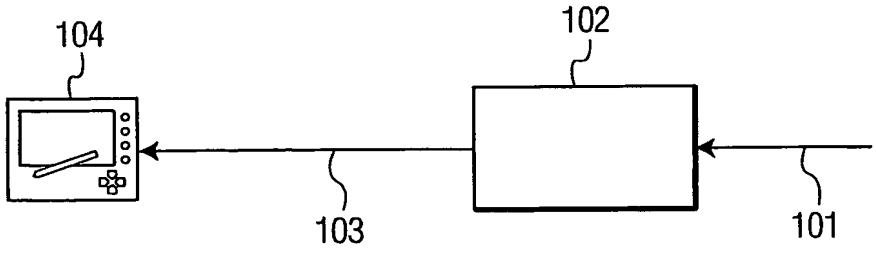

FIG. 1A

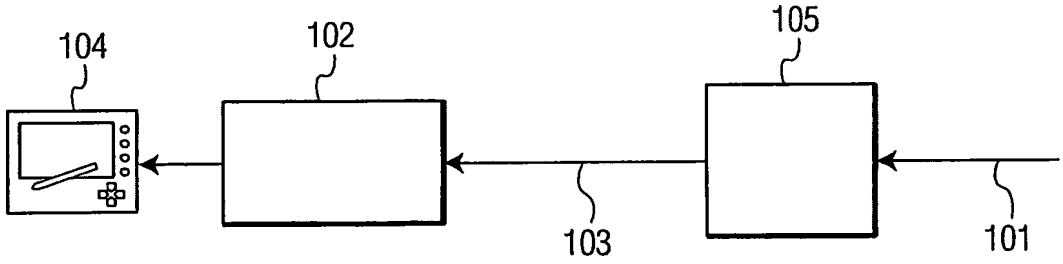

FIG. 1B

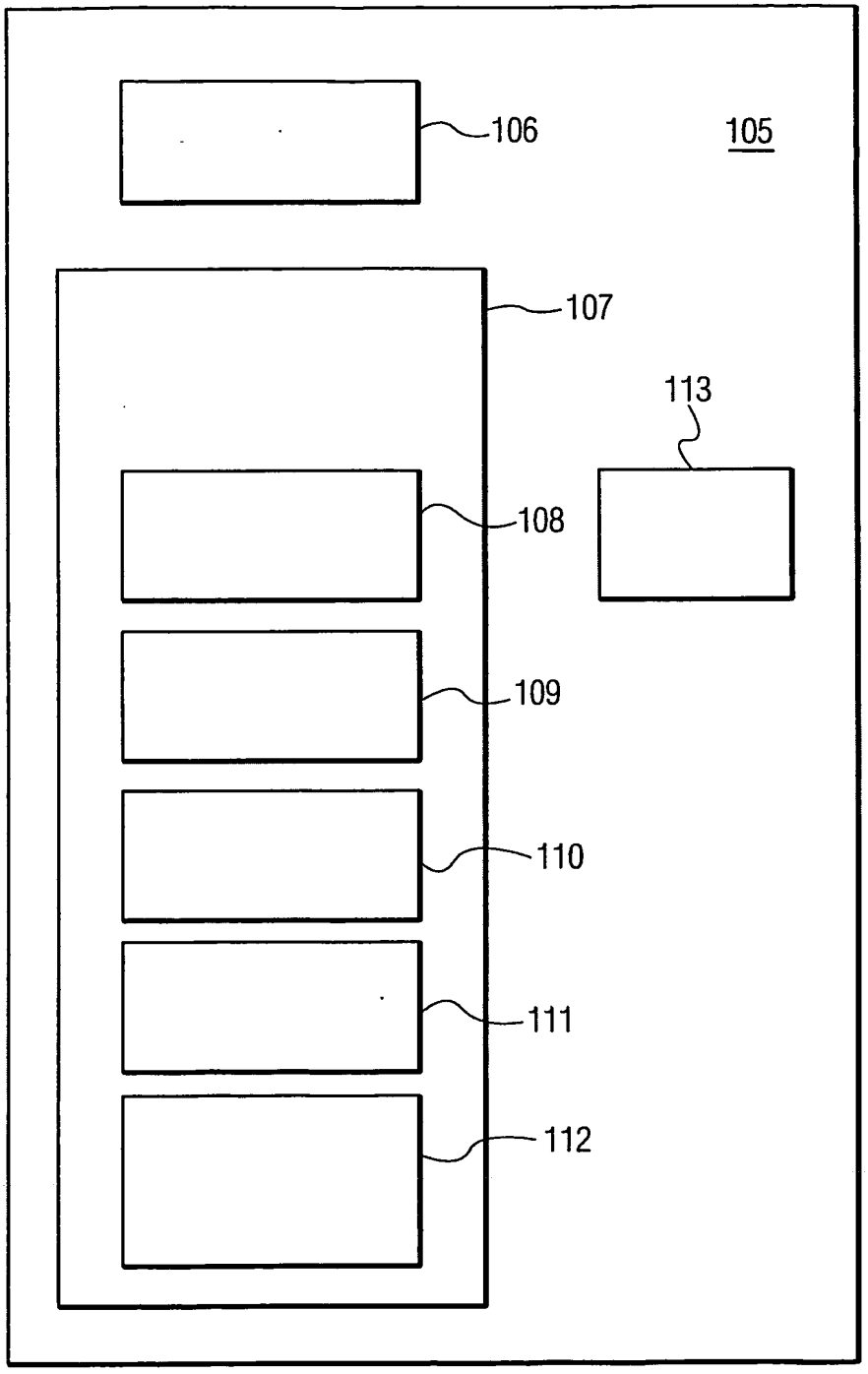

FIG. 1C

FIG. 2

FIG. 3

FIG. 4

FIG. 5

FIG. 6

FIG. 7

FIG. 8

FIG. 9

FIG. 10

FIG. 11

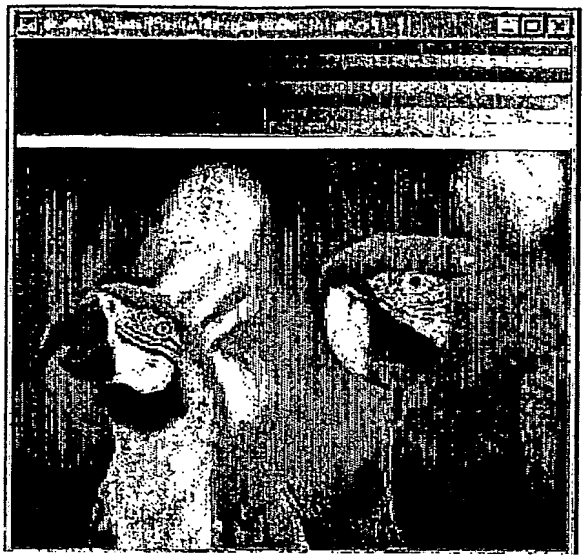

FIG. 12

FIG. 13

```
% Assume input image is 0<(R,G,B)<1

% Constants
MIN = 0                          (typically 0 or 16/255)
MAX = 1                          (typically 1 or 235/255)

IMAXFAC = g(A1,A2)               (reduce overall brightness, depends on input gamut and display gamut, e.g. 0.85)

% Effect size (larger with larger values of R,G,B)
ALPHA = f(R,G,B)  (e.g. max(R,G,B) or √(R^2+G^2+B^2), or c1R+c2G+c3B)


% Reduce overall brightness
(R,G,B) = (R,G,B)*IMAXFAC

% Gamma correction (go to linear color domain, this
% can be omitted for simplification, to reduce the
% number of operations, but the color transformations
% are no longer very accurate)
(R,G,B) = (R,G,B)^GAMMA

% Gamut correction (expansion)
(R,G,B) = inv(A2) * A1 * (R,G,B)

% Clipping values < MIN if any((R,G,B)<MIN), Subtract
% minimum multiplied with certain factor depending on
% intensity
(R,G,B) = (R,G,B) +  ALPHA*(MIN – min(R,G,B,MIN))
end

% Clipping values > MAX if any((R,G,B)>MAX),
% Scale RGB vector within maximum
(R,G,B) = (R,G,B) * (MAX/max(R,G,B,MAX))
end

% Inverse gamma correction, if needed
(R,G,B) = (R,G,B)^(1/GAMMA)
```

FIG. 14

```
% Assume input image is 0<(R,G,B)<1

% Constants
MIN = 0          (typically 0 or 16/255)
MAX = 1          (typically 1 or 235/255)
IMAXFAC = g(A1,A2)    (reduce overall brightness, depends on input gamut and display gamut, e.g. 0.85)

% Effect size (larger with larger values of R,G,B)
ALPHA = f(R,G,B)      (e.g. max(R,G,B) or √(R^2+G^2+B^2), or c1R+c2G+c3B)


% Reduce overall brightness
~~(R,G,B) = (R,G,B)*IMAXFAC~~
(R,G,B) = (R,G,B)*F(S);
% Where S is the saturation, that can be calculated from R,G,B
% and F(S) is a function of S that is equal to MAX for S=0 and
% equal to IMAXFAC for S = some (constant) value between 0 and
% the maximum saturation of the primary colors. The function F
% can be any continuously decreasing function, e.g. linear or
% cos^2. For the example below S=0.75 is the cut-off
% saturation for which F(S) is IMAXFAC.

% Gamma correction (go to linear color domain, this can be
% omitted for simplification, to reduce the number of
% operations, but the color transformations are no longer very
% accurate)
(R,G,B) = (R,G,B)^GAMMA

% Gamut correction (expansion)
(R,G,B) = inv(A2) * A1 * (R,G,B)

% Clipping values < MIN if any((R,G,B)<MIN),
% Subtract minimum multiplied with certain factor depending on
% intensity
(R,G,B) = 13R,G,B) +  ALPHA*(MIN – min(R,G,B,MIN))
end

% Clipping values > MAX if any((R,G,B)>MAX),
% Scale RGB vector within maximum
(R,G,B) = (R,G,B) * (MAX/max(R,G,B,MAX))
end

% Inverse gamma correction if needed
(R,G,B) = (R,G,B)^(1/GAMMA)

% Output image is between: MIN<(R,G,B)<MAX
```

FIG. 15

FIG. 16A

FIG. 16B

FIG. 16C

FIG. 16D

```
% Assume input image is 0<(R,G,B)<1
% Constants
MIN = 0          (typically 0 or 16/255)
MAX = 1          (typically 1 or 235/255)
IMAXFAC = g(A1,A2)    (reduce overall brightness, depends on input gamut and display gamut, e.g. 0.85)
% Effect size (larger with larger values of R,G,B)
ALPHA = f(R,G,B)      (e.g. max(R,G,B) or √(R^2+G^2+B^2), or c1R+c2G+c3B)
% Reduce overall brightness
(X)   [0.4306   0.3415   0.1784]  (R)
(Y) = [0.2220   0.7067   0.0713] * (G)
(Z)   [0.0202   0.1296   0.9393]  (B)
u' = 4*X/(X + 15*Y + 3*Z)
v' = 9*Y/(X + 15*Y + 3*Z)
S = 13*sqrt((u'-0.1978).^2+(v'-0.4683).^2);
SCUT = 0.75;
if S>SCUT,
  F = IMAXFAC;
else
  F = (((1+cos(S/SCUT*pi))/2).^2) *(1-IMAXFAC) + IMAXFAC;
end
(R,G,B) = (R,G,B)*F;
% Where S is the saturation, that can be calculated from R,G,B
% and F(S) is a function of S  that is equal to MAX for S=0 and
% equal to IMAXFAC for S = some (constant) value between 0 and
% the maximum saturation of the primary colors. The function F
% can be any continuously decreasing function, e.g. linear or
% cos^2. For the example below S=0.75 is the cut-off
% saturation for which F(S) is IMAXFAC.
% Gamma correction (go to linear color domain, this can be
% omitted for simplification, to reduce the number of
% operations, but the color transformations are no longer very
% accurate)
(R,G,B) = (R,G,B)^GAMMA
% Gamut correction (expansion)
(R,G,B) = inv(A2) * A1 * (R,G,B)
% Clipping values < MIN if any((R,G,B)<MIN),
% Subtract minimum multiplied with certain factor depending on
% intensity
(R,G,B) = (R,G,B) +  ALPHA*(MIN – min(R,G,B,MIN))
end
% Clipping values > MAX if any((R,G,B)>MAX),
% Scale RGB vector within maximum
(R,G,B) = (R,G,B) * (MAX/max(R,G,B,MAX))
end
% Inverse gamma correction if needed
(R,G,B) = (R,G,B)^(1/GAMMA)
% Output image is between: MIN<(R,G,B)<MAX
```

FIG. 17